Investigating the Effect of Damage Progression Model Choice on Prognostics Performance

Matthew Daigle [1] Indranil Roychoudhury [2] Sriram Narasimhan [1] Sankalita Saha [3] Bhaskar Saha [3] and Kai Goebel [4]

[1] *University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, CA, 94035, USA*
matthew.j.daigle@nasa.gov, sriram.narasimhan-1@nasa.gov
[2] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
indranil.roychoudhury@nasa.gov
[3] *MCT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
bhaskar.saha@nasa.gov, sankalita.saha-1@nasa.gov
[4] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

The success of model-based approaches to systems health management depends largely on the quality of the underlying models. In model-based prognostics, it is especially the quality of the damage progression models, i.e., the models describing how damage evolves as the system operates, that determines the accuracy and precision of remaining useful life predictions. Several common forms of these models are generally assumed in the literature, but are often not supported by physical evidence or physics-based analysis. In this paper, using a centrifugal pump as a case study, we develop different damage progression models. In simulation, we investigate how model changes influence prognostics performance. Results demonstrate that, in some cases, simple damage progression models are sufficient. But, in general, the results show a clear need for damage progression models that are accurate over long time horizons under varied loading conditions.

1. INTRODUCTION

Model-based prognostics is rooted in the use of models that describe the behavior of systems and components and how that behavior changes as wear and damage processes occur (Luo, Pattipati, Qiao, & Chigusa, 2008; Saha & Goebel, 2009; Daigle & Goebel, 2011). The problem of model-based prognostics fundamentally consists of two sequential problems, (*i*) a joint state-parameter estimation problem, in which, using the model, the health of a system or component is determined based on its observations; and (*ii*) a prediction problem, in which, using the model, the state-parameter distribution is simulated forward in time to compute *end of life* (EOL) and *remaining useful life* (RUL). The model must describe both how damage manifests in the system observations, and how damage progresses in time. Clearly, the prognostics performance inherently depends on the quality of the models used by the algorithms.

In modeling the complex engineering systems targeted by prognostics algorithms, many modeling choices must be made. In particular, one must decide on the appropriate level of abstraction at which to model the system in order to estimate system health and predict remaining life. The choice is mainly one of model granularity, i.e., the extent to which the model is broken down into parts, either structural or behavioral. The selected models must then provide enough fidelity to meet the prognostics performance requirements. But, model development cost, available level of expertise, model validation effort, and computational complexity all constrain the models that may be developed. For example, finer-grained models may result in increased model fidelity and thus increased prognostics performance, but may take more effort to construct and increase computational complexity. Therefore, a clear need exists to investigate the impact of such modeling choices on prognostics performance.

In this paper, we use a centrifugal pump as a case study with which to explore the impact of model quality on prognostics performance. Typically, developing a reliable model of nominal system operation is relatively straightforward, as the dynamics are usually well-understood in terms of first principles or physics equations, and, most importantly, there is typically sufficient data available with which to validate this model. The major difficulty lies in developing models of damage progression, because these models are often component-dependent, and so the understanding of these processes is generally lacking. Further, the data necessary to properly validate these models are, in practice, rarely available. Using the pump model, we develop several damage progression models and evaluate their effect on prognostics performance using simulation-based experiments. To the best of our knowledge, this, along with a companion paper exploring these issues

with application to battery health management (Saha, Quach, & Goebel, 2011), is the first time this type of analysis has been performed within the context of prognostics.

The paper is organized as follows. Section 2 describes the model-based prognostics framework. Section 3 presents the modeling methodology and develops the centrifugal pump model with several damage progression models. Section 4 generalizes the different models within the framework of model abstraction. Section 5 describes the particle filter-based damage estimation method, and Section 6 discusses the prediction methodology. Section 7 provides results from a number of simulation-based experiments and evaluates the effect of the different damage progression models on prognostics performance. Section 8 concludes the paper.

2. MODEL-BASED PROGNOSTICS

We assume the system model may be described using

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t))$$
$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)),$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$ is the parameter vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(t) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation. The model may be nonlinear with no restrictions on the functional forms of $\mathbf{f}$ or $\mathbf{h}$, and the noise terms may be nonlinearly coupled with the states and parameters. The parameters $\boldsymbol{\theta}(t)$ evolve in an unknown way.

The goal of prognostics is to predict EOL (and/or RUL) at a given time point t_P using the discrete sequence of observations up to time t_P, denoted as $\mathbf{y}_{0:t_P}$. EOL is defined as the time point at which the component no longer meets a functional or performance requirement. In general, these requirements do not need to be directly tied to permanent failure, rather, they refer to a state of the system that is undesirable. The system can leave this state through repair or other actions, and sometimes no action is needed and the component needs only to rest (e.g., with power electronics, or self-recharge of batteries). These functional requirements may be expressed through a threshold, beyond which the component is considered to have failed. In general, we may express this threshold as a function of the system state and parameters, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t))$, where $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ if a requirement is violated, and 0 otherwise.

So, EOL may be defined as

$$EOL(t_P) \triangleq \inf\{t \in \mathbb{R} : t \geq t_P \wedge T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1\},$$

i.e., EOL is the earliest time point at which the threshold is reached. RUL may then be defined with

$$RUL(t_P) \triangleq EOL(t_P) - t_P.$$

Due to various sources of uncertainty, including uncertainty in the model, the goal is to compute a probability distribution of the EOL or RUL. We compute, at time t_P, $p(EOL(t_p)|\mathbf{y}_{0:t_P})$ or $p(RUL(t_P)|\mathbf{y}_{0:t_P})$.

In model-based prognostics, there are two fundamental problems: (*i*) joint state-parameter estimation, and (*ii*) prediction. In discrete time k, we estimate $\mathbf{x}_k$ and $\boldsymbol{\theta}_k$, and use these estimates to predict EOL and RUL at desired time points. The model-based prognostics architecture is shown in Fig. 1 (Daigle & Goebel, 2011). Given inputs $\mathbf{u}_k$, the system provides measured outputs $\mathbf{y}_k$. If available, a fault detection, isolation, and identification (FDII) module may be used to determine which damage mechanisms are active, represented as a fault set $\mathbf{F}$. The damage estimation module may use this result to limit the dimension of the estimation problem. It determines estimates of the states and unknown parameters, represented as a probability distribution $p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{0:k})$. The prediction module uses the joint state-parameter distribution, along with hypothesized future inputs, to compute EOL and RUL as probability distributions $p(EOL_{k_P}|\mathbf{y}_{0:k_P})$ and $p(RUL_{k_P}|\mathbf{y}_{0:k_P})$ at given prediction times k_P. In this paper, we assume a solution to FDII that provides us with the single active damage mechanism, initiating prognostics.

Prognostics performance is evaluated based on the accuracy and precision of the predictions. We use the relative accuracy (RA) metric (Saxena, Celaya, Saha, Saha, & Goebel, 2010) to characterize prediction accuracy. For a given prediction time k_P, RA is defined as

$$\text{RA}_{k_P} = 100\left(1 - \frac{|RUL^*_{k_P} - \widehat{RUL}_{k_P}|}{RUL^*_{k_P}}\right),$$

where $RUL^*_{k_P}$ is the true RUL at time k_P, and $\widehat{RUL}_{k_P}$ is the mean of the prediction. The prognostic horizon (PH) refers to the time between EOL and the first prediction that meets some accuracy requirement RA^* (e.g., 90%):

$$PH = 100\frac{EOL^* - \min\{k_P : RA_{k_P} \geq RA^*\}}{EOL^*},$$

where EOL^* denotes the true EOL. A larger value means an accurate prediction is available earlier. This is a version of the PH metric given in (Saxena et al., 2010) normalized to EOL. Prediction spread is computed using relative median absolute deviation (RMAD):

$$\text{RMAD}(X) = 100\frac{\text{Median}_i\left(|X_i - \text{Median}_j(X_j)|\right)}{\text{Median}_j(X_j)},$$

where X is a data set and X_i is an element of that set.

3. PUMP MODELING

In our modeling methodology, we first describe a nominal model of system behavior. We then extend the model by including *damage progression functions* within the state equation $\mathbf{f}$ that describe how *damage variables* $\mathbf{d}(t) \subseteq \mathbf{x}(t)$ evolve over time. The damage progression functions are parameterized by unknown *wear parameters* $\mathbf{w}(t) \subseteq \boldsymbol{\theta}(t)$. We use

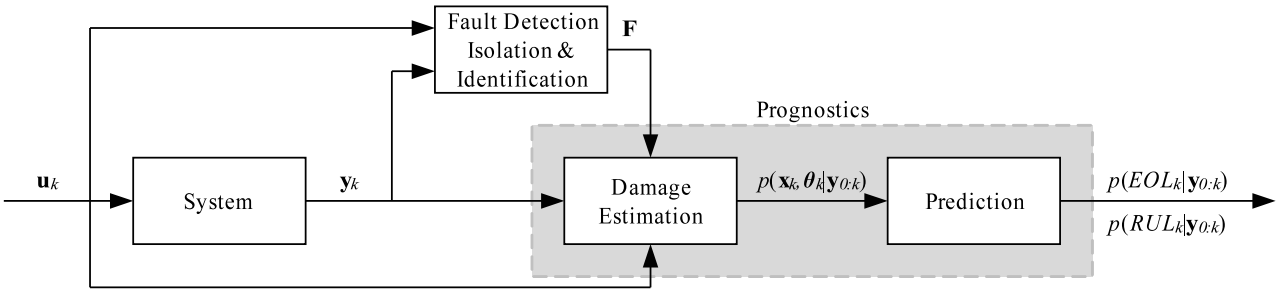


Figure 1. Prognostics architecture.

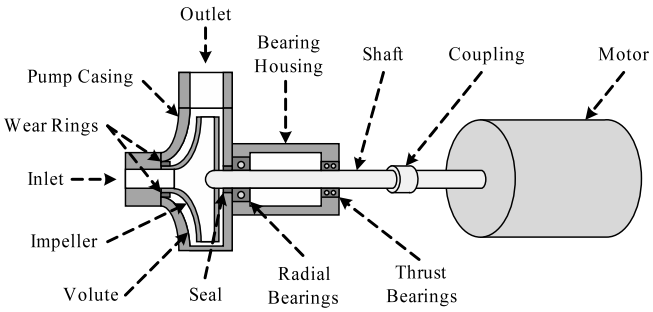


Figure 2. Centrifugal pump.

a centrifugal pump as a case study. In this section, we first describe the nominal model of the pump, and then describe common damage progression models.

3.1 Nominal Model

A schematic of a typical centrifugal pump is shown in Fig. 2. Fluid enters the inlet, and the rotation of the impeller, driven by an electric motor, forces fluid through the outlet. The radial and thrust bearings help to minimize friction along the pump shaft. The bearing housing contains oil which lubricates the bearings. A seal prevents fluid flow into the bearing housing. Wear rings prevent internal pump leakage from the outlet to the inlet side of the impeller, but a small clearance is typically allowed to minimize friction. The nominal pump model has been described previously in (Daigle & Goebel, 2011), and we review it here for completeness.

The state of the pump is given by

$$\mathbf{x}(t) = \begin{bmatrix} \omega(t) & T_t(t) & T_r(t) & T_o(t) \end{bmatrix}^T,$$

where $\omega(t)$ is the rotational velocity of the pump, $T_t(t)$ is the thrust bearing temperature, $T_r(t)$ is the radial bearing temperature, and $T_o(t)$ is the oil temperature.

The rotational velocity of the pump is described using a torque balance,

$$\dot{\omega} = \frac{1}{J}\left(\tau_e(t) - r\omega(t) - \tau_L(t)\right),$$

where J is the lumped motor/pump inertia, τ_e is the electromagnetic torque provided by the motor, r is the lumped friction parameter, and τ_L is the load torque. In an induction motor, a voltage is applied to the stator, which creates a current through the stator coils. A polyphase voltage applied to the stator creates a rotating magnetic field that induces a current in the rotor, causing it to turn. The torque produced on the rotor is nonzero only when there is a difference between the synchronous speed of the supply voltage, ω_s and the mechanical rotation, ω. This *slip* is defined as

$$s = \frac{\omega_s - \omega}{\omega_s}.$$

The expression for the torque τ_e is derived from an equivalent circuit representation for the three-phase induction motor based on rotor and stator resistances and inductances, and the slip s (Lyshevski, 1999):

$$\tau_e = \frac{npR_2}{s\omega_s} \frac{V_{rms}^2}{(R_1 + R_2/s)^2 + (\omega_s L_1 + \omega_s L_2)^2},$$

where R_1 is the stator resistance, L_1 is the stator inductance, R_2 is the rotor resistance, L_2 is the rotor inductance, n is the number of phases (typically 3), and p is the number of magnetic pole pairs. The dependence of torque on slip creates a feedback loop that causes the rotor to follow the rotation of the magnetic field. The rotor speed may be controlled by changing the input frequency ω_s.

The load torque τ_L is a polynomial function of the pump flow rate and the impeller rotational velocity (Wolfram, Fussel, Brune, & Isermann, 2001; Kallesøe, 2005):

$$\tau_L = a_0\omega^2 + a_1\omega Q - a_2 Q^2,$$

where Q is the flow, and a_0, a_1, and a_2 are coefficients derived from the pump geometry (Kallesøe, 2005).

The rotation of the impeller creates a pressure difference from the inlet to the outlet of the pump, which drives the pump flow, Q. The pump pressure is computed as

$$p_p = A\omega^2 + b_1\omega Q - b_2 Q^2,$$

where A is the impeller area, and b_1 and b_2 are coefficients derived from the pump geometry. The discharge flow, Q, is comprised of the flow through the impeller, Q_i, and a leakage flow, Q_l:

$$Q = Q_i - Q_l.$$

The flow through the impeller is computed using the pressure differences:

$$Q_i = c\sqrt{|p_s + p_p - p_d|}sign(p_s + p_p - p_d),$$

where c is a flow coefficient, p_s is the suction pressure, and p_d is the discharge pressure. The small (normal) leakage flow from the discharge end to the suction end due to the clearance between the wear rings and the impeller is described by

$$Q_l = c_l\sqrt{|p_d - p_s|}sign(p_d - p_s),$$

where c_l is a flow coefficient.

Pump temperatures are often monitored as indicators of pump condition. The oil heats up due to the radial and thrust bearings and cools to the environment:

$$\dot{T}_o = \frac{1}{J_o}(H_{o,1}(T_t - T_o) + H_{o,2}(T_r - T_o) - H_{o,3}(T_o - T_a)),$$

where J_o is the thermal inertia of the oil, and the $H_{o,i}$ terms are heat transfer coefficients. The thrust bearings heat up due to the friction between the pump shaft and the bearings, and cool to the oil and the environment:

$$\dot{T}_t = \frac{1}{J_t}(r_t\omega^2 - H_{t,1}(T_t - T_o) - H_{t,2}(T_t - T_a)),$$

where J_t is the thermal inertia of the thrust bearings, r_t is the friction coefficient for the thrust bearings, and the $H_{t,i}$ terms are heat transfer coefficients. The radial bearings behave similarly:

$$\dot{T}_r = \frac{1}{J_r}(r_r\omega^2 - H_{r,1}(T_r - T_o) - H_{r,2}(T_r - T_a))$$

where J_r is the thermal inertia of the radial bearings, r_r is the friction coefficient for the radial bearings, and the $H_{r,i}$ terms are heat transfer coefficients.

The overall input vector $\mathbf{u}$ is given by

$$\mathbf{u}(t) = \begin{bmatrix} p_s(t) & p_d(t) & T_a(t) & V(t) & \omega_s(t) \end{bmatrix}^T.$$

The measurement vector $\mathbf{y}$ is given by

$$\mathbf{y}(t) = \begin{bmatrix} \omega(t) & Q(t) & T_t(t) & T_r(t) & T_o(t) \end{bmatrix}^T.$$

Fig. 3 shows nominal pump operation. Input voltage and line frequency are varied to control the pump speed. Initially, slip is 1, and this produces an electromagnetic torque that causes the rotation of the motor to match the rotation of the magnetic field, with a small amount of slip remaining (depending on the load). Fluid flows through the pump due to the impeller rotation. The bearings heat and cool as the pump rotation increases and decreases.

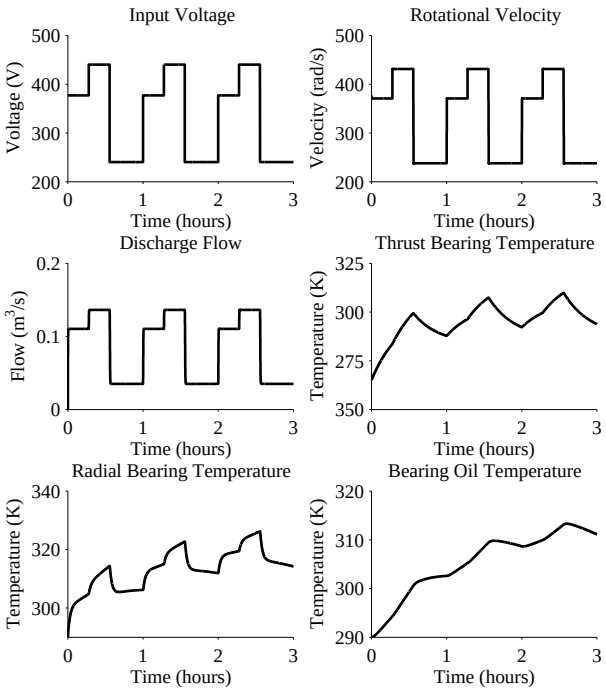


Figure 3. Nominal pump operation.

3.2 Damage Modeling

The most significant forms of damage for pumps are impeller wear, caused by cavitation and erosion by the flow, and bearing failure, caused by friction-induced wear of the bearings. In each case, we map the damage to a particular parameter in the nominal model, and this parameter becomes a damage variable in $\mathbf{d}(t)$ that evolves by a damage progression function. Several types of damage progression models have been explored in literature. In this paper, we focus on macro-level, lumped-parameter models. Within this modeling style, damage evolves as a function of dynamic energy-related variables. Several common forms may be assumed here, including linear, polynomial, and exponential, as these forms have been observed in practice. We derive these forms for the considered damage modes as well as wear-based models based on physics analysis.

Impeller wear is represented as a decrease in impeller area A (Biswas & Mahadevan, 2007; Tu et al., 2007; Daigle & Goebel, 2011). Impeller wear can only progress when flow through the impeller, Q_i, is nonzero. So, the rate of change of impeller area, $\dot{A}$, must be a function of Q_i. We consider the following damage progression models based on the common observed forms:

$$\dot{A} = -w_A Q_i \quad (1)$$

$$\dot{A} = -w_A Q_i^2 \quad (2)$$

$$\dot{A} = -w_{A1} Q_i - w_{A2} Q^2 \quad (3)$$

$$\dot{A} = -w_{A1} \exp(w_{A2} Q_i), \quad (4)$$

where w_A, w_{A1}, and w_{A2} are unknown wear parameters.

From a physics analysis, we see that the erosive wear equation applies here (Hutchings, 1992). The erosive wear rate is proportional to fluid velocity times friction force. Fluid velocity is proportional to volumetric flow rate, and friction force is proportional to fluid velocity, so, lumping the proportionality constants into the wear coefficient w_A, we obtain

$$\dot{A} = -w_A Q_i^2. \tag{5}$$

Note that this agrees with one of the commonly assumed damage forms, equation 2, above.

A decrease in the impeller area will decrease the pump pressure, which, in turn, reduces the delivered flow, and, therefore, pump efficiency. The pump must operate at a certain minimal efficiency. This requirement defines an EOL criteria. We define A^- as the minimum value of the impeller area at which this requirement is met, hence, $T_{EOL} = 1$ if $A(t) < A^-$.

The damage progression up to EOL for impeller wear is shown in Fig. 4a for equation 5, for the rotational velocity alternating between 3600 RPM for the first half of every hour of usage and 4300 RPM for the second half, causing the pump flow to alternate as well. Within a given cycle, shown in the inset of Fig. 4a, the damage progresses at two different rates, but over a long time horizon, the damage progression appears fairly linear. This suggests that a linear approximation may suffice for accurate long-term predictions if the future inputs cycle in the same way. The damage progression rate actually decreases slightly over time, because as impeller area decreases, flow will decrease, and therefore $\dot{A}$ will diminish.

Bearing wear is captured as an increase in the corresponding friction coefficient (Daigle & Goebel, 2011). Bearing wear can only occur when the pump is rotating, i.e., ω is nonzero. So, the rate of change of the bearing friction coefficient, $\dot{r}_t$ for the thrust bearing, and $\dot{r}_r$ for the radial bearing, must be a function of ω. For the thrust bearing wear, we consider the following damage progression models based on the common observed forms:

$$\dot{r}_t(t) = w_t\omega \tag{6}$$
$$\dot{r}_t(t) = w_t\omega^2 \tag{7}$$
$$\dot{r}_t(t) = w_{t1}\omega + w_{t2}\omega^2 \tag{8}$$
$$\dot{r}_t(t) = w_{t1}\exp(w_{t2}\omega), \tag{9}$$

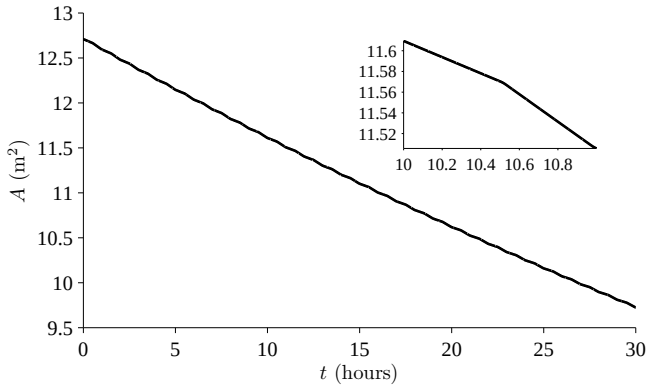
where w_t, w_{t1}, and w_{t2} are unknown wear parameters. For the radial bearing, the equations are the same, but with the t subscript replaced by an r subscript:

$$\dot{r}_r(t) = w_r\omega \tag{10}$$
$$\dot{r}_r(t) = w_r\omega^2 \tag{11}$$
$$\dot{r}_r(t) = w_{r1}\omega + w_{r2}\omega^2 \tag{12}$$
$$\dot{r}_r(t) = w_{r1}\exp(w_{r2}\omega). \tag{13}$$

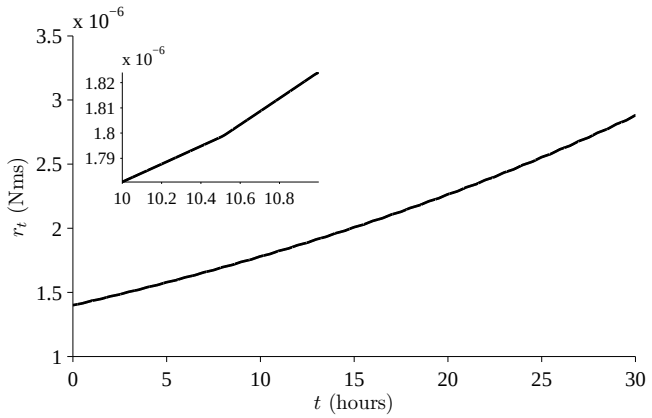
From a physics analysis, we observe that sliding and rolling friction generate wear of material which increases the coefficient of friction (Hutchings, 1992; Daigle & Goebel, 2010):

$$\dot{r}_t(t) = w_t r_t \omega^2 \tag{14}$$
$$\dot{r}_r(t) = w_r r_r \omega^2, \tag{15}$$

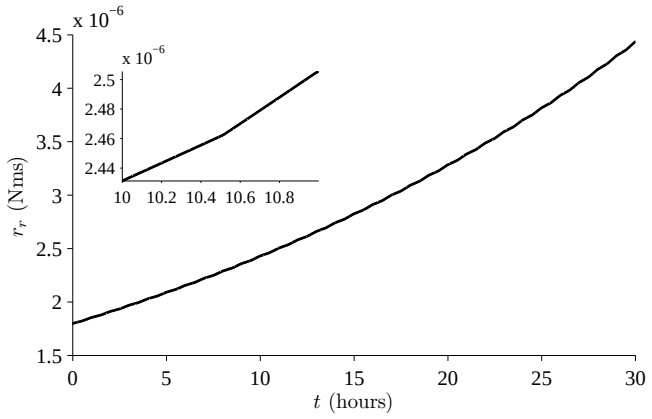
where w_t and w_r are the wear parameters. Note that equations 6–9 neglect the direct relationship between $\dot{r}_t$ and r_t.



(a) Damage progression for impeller wear.



(b) Damage progression for thrust bearing wear.



(c) Damage progression for radial bearing wear.

Figure 4. Damage progression for the pump.

Changes in bearing friction can be observed by means of the bearing temperatures. Limits on the maximum values of these temperatures define EOL for bearing wear. We define r_t^+ and r_r^+ as the maximum permissible values of the friction coefficients, before the temperature limits are exceeded over a typical usage cycle. So, $T_{EOL} = 1$ if $r_t(t) > r_t^+$ or $r_r(t) > r_r^+$.

Damage progression up to EOL for bearing wear is shown in Figs. 4b and 4c, for equations 14 and 15, with the rotational velocity again alternating between 3600 RPM and 4300 RPM. In this case, the rate of damage progression increases over time. Therefore, a simple linear approximation would not be accurate. This behavior occurs because $\dot{r}_t(t)$ increases with $r_t(t)$, and $\dot{r}_r(t)$ increases with $r_r(t)$.

4. Model Abstraction

The previous section presented a number of different models. In general, these differences may be captured by the idea of *model abstraction* (Frantz, 1995; Lee & Fishwick, 1996; Zeigler, Praehofer, & Kim, 2000). Abstraction is driven by the questions that the model must address. For prognostics, the models must address the question of the EOL/RUL of a system. In order to do this, the models must (*i*) describe how damage manifests in the system outputs (i.e., measured variables or computed features), so that damage estimation can be performed; and (*ii*) describe how damage evolves in time as a function of the system loading, so that prediction can be performed. The chosen level of model abstraction must be such that these tasks can be accomplished at the desired level of performance.

Abstraction is a process of simplification. Common abstractions include aggregation, omission, linearization, deterministic/stochastic replacement, and formalism transformation (e.g., differential equations to discrete-event systems) (Zeigler et al., 2000). These abstractions may manifest as *structural abstraction*, in which the model is abstracted by its structure, or *behavioral abstraction*, in which the model is abstracted by its behaviors (Lee & Fishwick, 1996). For example, a structural abstraction might ignore the individual circuit elements of an electric motor and aggregate them into a lumped component. A behavioral abstraction might omit the individual processes and effects comprising a damage progression process and instead consider their lumped effects. Or, perhaps a given process might really take on an exponential form, but is abstracted to a linear form. The linear form consists of a simpler relationship that is described by fewer free parameters.

Model granularity is a particular measure of model abstraction. The *granularity* of a model is the extent to which it is divided into smaller parts. The concept of granularity does not address the degree of complexity of the specific functional relationships within a part of the model. Granularity can manifest both structurally and behaviorally. For example, a lumped parameter model is coarser-grained than a finite element model. In the context of physics-based prognostics models, a model with fine granularity may include more lower-level physical processes (e.g., micro-level effects rather than macro-level effects), or model processes at a greater level of detail, than a model with coarse granularity.

In quantifiable terms, granularity may be expressed using the number of state variables, the number of relationships between them, and the number of free (unknown) parameters. By definition, the state variables are the minimal set of variables needed to describe the state of the system as it progresses through time. So a finer-grained model may entail an additional number of state variables because aspects of the physical description that were not captured before are now described. With the same state variables, a model may also become more granular by adding functional relationships between the state variables. In a linear system, with $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, this would correspond to zeros in the $\mathbf{A}$ matrix becoming nonzero. Note that this is only a fair comparison between two models capturing the same process.

The different damage models developed in Section 3.2 can be viewed within this framework. For a particular damage mode, the different damage models each capture the same physical process, i.e., the damage progression, but make different assumptions about the complexity of the process. Thus, these models capture damage progression at different levels of behavioral abstraction. For example, for the impeller wear, the polynomial form (equation 3) may be viewed as less abstract than both the linear (equation 1) and squared forms (equation 2), because it is a sum of these individual processes. For the bearing wear, equations 6–9 are all coarser-grained models than 14, because they neglect the direct relationship between $\dot{r}_t$ and r_t.

One may describe the system behavior in very low-level physical relationships, but, of course, there are trade-offs to be made among the modeling constraints. A finer-grained model takes more effort to develop and validate, and may result in an increased computational cost. It also may result in an increase in the number of free parameters, which increases the complexity of the joint state-parameter estimation problem. The increase in model development cost to create models with finer granularity is justified only when it results in an appropriate increase in fidelity (i.e., the extent to which a model reproduces the observable behaviors of the system being modeled) and a corresponding increase in prognostics performance. Also, higher levels of abstraction make sense when the computation associated with lower levels of abstraction becomes too complicated for practical implementation. Requirements on prognostics performance and constraints on model size, development cost, level of modeling expertise, and computational complexity all drive the model development process.

5. DAMAGE ESTIMATION

Damage estimation is fundamentally a joint state-parameter estimation problem, i.e., computation of $p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{0:k})$. The damage states and wear parameters must be estimated along with the other state variables and unknown parameters of the system. We use the *particle filter* (Arulampalam, Maskell, Gordon, & Clapp, 2002) as a general solution to this problem.

In a particle filter, the state distribution is approximated by a set of discrete weighted samples, or *particles*:

$$\{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i), w_k^i\}_{i=1}^N,$$

where N denotes the number of particles, and for particle i, $\mathbf{x}_k^i$ denotes the state vector estimate, $\boldsymbol{\theta}_k^i$ denotes the parameter vector estimate, and w_k^i denotes the weight. The posterior density is approximated by

$$p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{0:k}) \approx \sum_{i=1}^{N} w_k^i \delta_{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i)}(d\mathbf{x}_k d\boldsymbol{\theta}_k),$$

where $\delta_{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i)}(d\mathbf{x}_k d\boldsymbol{\theta}_k)$ denotes the Dirac delta function located at $(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i)$.

We use the sampling importance resampling (SIR) particle filter. Each particle is propagated forward to time k by first sampling new parameter values, and then sampling new states using the model. The particle weight is assigned using $\mathbf{y}_k$. The weights are then normalized, followed by the resampling step. Pseudocode is given in (Arulampalam et al., 2002; Daigle & Goebel, 2011).

Parameter values are sampled using a random walk, i.e., for parameter θ, $\theta_k = \theta_{k-1} + \xi_{k-1}$, where ξ_{k-1} is sampled from some distribution. Particles generated with parameter values closest to the true values should be assigned higher weight and allow the particle filter to converge to the true values. The random walk variance is modified dynamically online to maintain a user-specified relative spread of the unknown wear parameters using the variance control algorithm presented in (Daigle & Goebel, 2011). The algorithm increases or decreases the random walk variance proportional to the difference between the desired spread and the actual spread, computed with relative median absolute deviation (RMAD). The algorithm behavior is specified using four parameters: the desired spread during the initial convergence period, v_0^* (e.g., 50%), the threshold that specifies the end of the convergence period, T (e.g., 60%), the final desired spread v_∞^* (e.g., 10%), and the proportional gain P (e.g. 1×10^{-3}). The spread is first controlled to v_0^* until the spread reaches T, at which point it is controlled to v_∞^*.

6. PREDICTION

Given the current joint state-parameter estimate at a desired prediction time k_P, $p(\mathbf{x}_{k_P}, \boldsymbol{\theta}_{k_P}|\mathbf{y}_{0:k_P})$, the prediction step computes $p(EOL_{k_P}|\mathbf{y}_{0:k_P})$ and $p(RUL_{k_P}|\mathbf{y}_{0:k_P})$. The particle filter provides

$$p(\mathbf{x}_{k_P}, \boldsymbol{\theta}_{k_P}|\mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w_{k_P}^i \delta_{(\mathbf{x}_{k_P}^i, \boldsymbol{\theta}_{k_P}^i)}(d\mathbf{x}_{k_P} d\boldsymbol{\theta}_{k_P}).$$

We approximate a prediction distribution n steps forward as (Doucet, Godsill, & Andrieu, 2000)

$$p(\mathbf{x}_{k_P+n}, \boldsymbol{\theta}_{k_P+n}|\mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w_{k_P}^i \delta_{(\mathbf{x}_{k_P+n}^i, \boldsymbol{\theta}_{k_P+n}^i)}(d\mathbf{x}_{k_P+n} d\boldsymbol{\theta}_{k_P+n}).$$

Similarly, we approximate the EOL as

$$p(EOL_{k_P}|\mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w_{k_P}^i \delta_{EOL_{k_P}^i}(dEOL_{k_P}).$$

To compute EOL, then, we propagate each particle forward to its own EOL and use that particle's weight at k_P for the weight of its EOL prediction. The prediction is made using hypothesized future inputs of the system. In this work, we assume these inputs are known in advance. Pseudocode for the prediction algorithm is given in (Daigle & Goebel, 2011).

7. RESULTS

We ran a number of simulation experiments for the different pump models in order to evaluate the relative performance. We took the damage models using the physics-based wear equations as the reference models that generated the measurement data. The model used by the prognostics algorithm was either the reference model $\mathcal{M}$ (using equations 5, 14, and 15), the linear model $\mathcal{M}_{Linear}$ (using equations 1, 6, and 10), the squared model $\mathcal{M}_{Squared}$ (using equations 2, 7, and 11), the second order polynomial model $\mathcal{M}_{Poly}$ (using equations 3, 8, and 12), or the exponential model $\mathcal{M}_{Exp}$ (using equations 4, 9, and 13). In each experiment, the pump speed cycled from 3600 RPM for the first half of every hour of usage to 4300 RPM for the second half hour.

In order to analyze results on a per-damage mode basis, in each experiment we assumed only a single damage mode was active. We selected the reference model's wear parameter values randomly in each experiment, within $[0.5 \times 10^{-3}, 4 \times 10^{-3}]$ for w_A, in $[0.5 \times 10^{-11}, 7 \times 10^{-11}]$ for w_t and w_r, such that the maximum wear rates corresponded to a minimum EOL of 20 hours. The particle filters had to estimate the states and the wear parameters associated with their assumed damage progression models. We considered the case where the future input was known in order to focus on the differences in performance based on the different assumed damage models. We also varied the process noise variance from 0, to nominal, and 10 times nominal, in order to artificially represent the nominal model at various levels of granularity.

Model	v	$\overline{\text{RA}}$	$\overline{\text{RMAD}}_{RUL}$
$\mathcal{M}$	0	97.87	10.33
	1	97.42	10.30
	10	97.63	10.41
$\mathcal{M}_{Linear}$	0	94.12	10.42
	1	92.28	10.91
	10	83.68	12.42
$\mathcal{M}_{Poly}$	0	97.55	3.35
	1	96.97	6.62
	10	89.98	10.55
$\mathcal{M}_{Exp}$	0	87.27	12.87
	1	88.83	13.01
	10	81.78	12.90

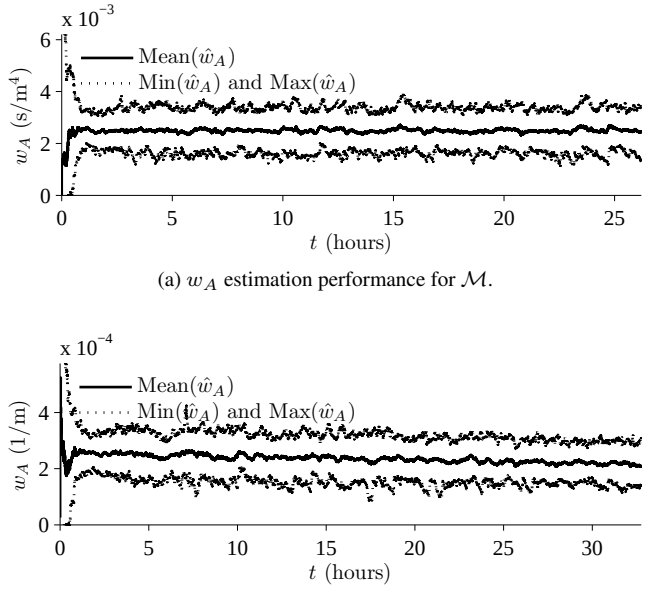
Table 1. Prognostics Performance for Impeller Wear

The assumption here is that the process noise represents finer-grained unmodeled processes that are not incorporated into the model and therefore look like noise.

Prognostics performance is dependent on both the underlying models used and on the prognostics algorithm. In order to focus on the dependence on modeling, we fix the algorithm and its parameters. The particle filter used $N = 500$ in all cases. The variance control algorithm used $v_0^* = 50\%$, $T = 60\%$, $v_\infty^* = 10\%$ in all cases, and used $P = 1 \times 10^{-3}$ for the damage models with one unknown wear parameter and $P = 1 \times 10^{-4}$ for those with two unknown wear parameters.

The prognostics performance results for impeller wear using different damage models and different levels of process noise variance are shown in Table 1. The process noise variance multiplier is shown in the second column of the table. We average RA over all prediction points to summarize the accuracy, denoted using $\overline{\text{RA}}$, and we average RMAD over all prediction points to summarize the spread, denoted using $\overline{\text{RMAD}}_{RUL}$. Multiple experiments were run for each case, and the table presents the averaged results. We can see that the linear damage model actually does fairly well. Its performance decreases as process noise increases, but for small amounts of process noise the accuracy is over 90%. The polynomial model also does well, which is expected since the second term by itself is the reference damage model. The particle filter still estimates a linear component which tracks damage progression over a short term fairly well, and it is the presence of this linear term that causes the accuracy to decrease. The exponential model does not do as well, partly because the behavior is very sensitive to the wear parameter inside the exponential function, w_{A2}, and so estimating both wear parameters simultaneously is more difficult for the particle filter.

The estimation performance using the reference model and the linear model is compared in Fig. 5. In both cases, the



(a) w_A estimation performance for $\mathcal{M}$.

(b) w_A estimation performance for $\mathcal{M}_{Linear}$.

Figure 5. Impeller wear parameter estimation.

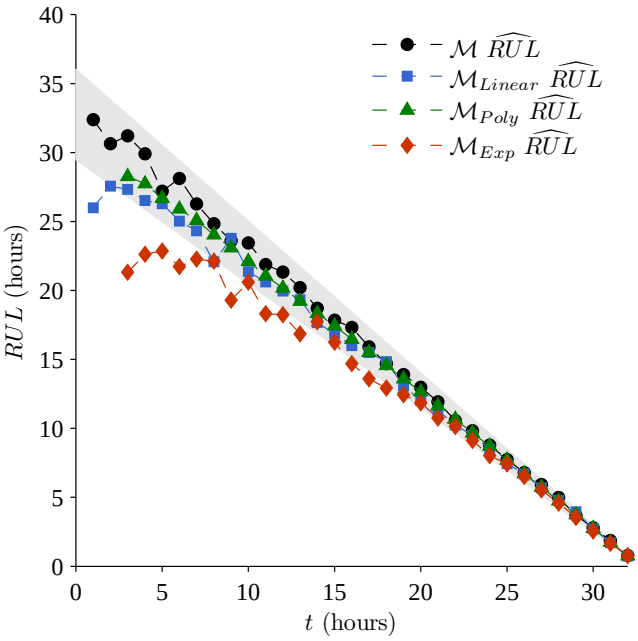


Figure 6. Impeller wear RUL prediction performance.

damage variable, A, was tracked well. When using the same damage model as in the reference model, the wear parameter is tracked easily and after convergence remains fairly constant. As a result, the predictions, shown in Fig. 6, using the mean, denoted by $\widehat{RUL}$, are very accurate and appear within 10% of the true value at all prediction points (shown using the gray cone in the figure). Because the rate of damage progres-

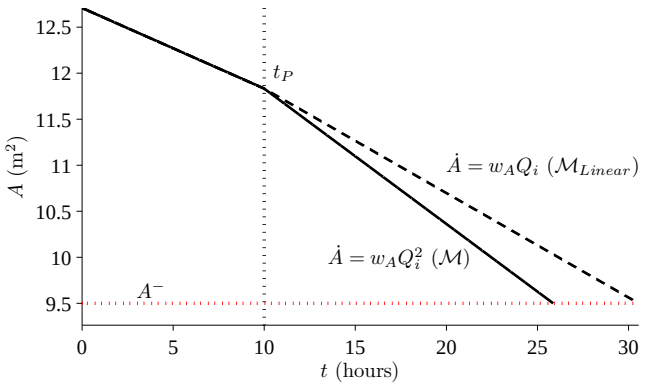


Figure 7. Impeller wear damage progression prediction, where at t_P, Q_i increases by 30%.

sion in the reference model decreases slowly over time, and the linear model does not accurately capture that behavior, its wear parameter estimate decreases over time in order to keep tracking the short-term damage progression. This is reflected also in the RUL predictions. Although the RUL accuracy is also very good, it is clear that it consistently underestimates the true RUL, because at any point in time it is overestimating the rate of damage progression that would occur in the future. However, the prognostic horizon is still very high. As shown in Fig. 6, by the second or third prediction, the predictions are all within the desired accuracy cone, except for the exponential model, which has PH of around 60%, meaning that at 60% life remaining, the exponential model is making accurate predictions. In many practical situations that may, in fact, be enough time for decision-making.

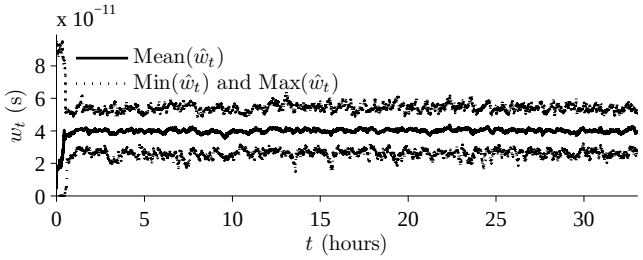
For impeller wear, the linear model does well in this case because the future loading is the same as the current loading. If Q_i is held constant, then the reference damage model $\dot{A} = w_A Q_i^2$, which equals $(w_A Q_i)Q_i$, looks exactly like the linear form because the product $w_A Q_i$ is constant. So the particle filter would estimate a wear parameter for the linear model that is the product of the wear parameter for the reference model multiplied by Q_i. So under constant loading, the linear model, or any other damage model that predicts a constant $\dot{A}$ under uniform loading, will produce accurate predictions. But, if the future loading is different than the current loading, then the product $w_A Q_i$ will change and the wear parameter estimated for the linear model will no longer be valid. This is illustrated in Fig. 7. At t_P, Q_i increases by 30%. The algorithm using the reference damage model captures the relationship between $\dot{A}$ and Q_i consistently with the simulation, and predicts EOL to be a little over 25 hours. In contrast, the linear model overestimates the RUL, because its wear parameter was tuned to the previous value of Q_i, and results in a RA of only around 80%. So for complex loading situations, it is important to correctly capture the relationship between loading and damage progression.

Model	v	$\overline{\text{RA}}$	$\overline{\text{RMAD}}_{RUL}$
$\mathcal{M}$	0	97.80	11.61
	1	97.57	11.43
	10	97.50	11.18
$\mathcal{M}_{Linear}$	0	79.93	10.72
	1	83.93	10.79
	10	82.45	9.41
$\mathcal{M}_{Squared}$	0	78.05	11.59
	1	79.68	12.15
	10	74.59	11.17
$\mathcal{M}_{Poly}$	0	78.43	6.07
	1	78.94	9.09
	10	76.48	11.76
$\mathcal{M}_{Exp}$	0	82.34	9.23
	1	79.87	12.43
	10	69.37	21.32

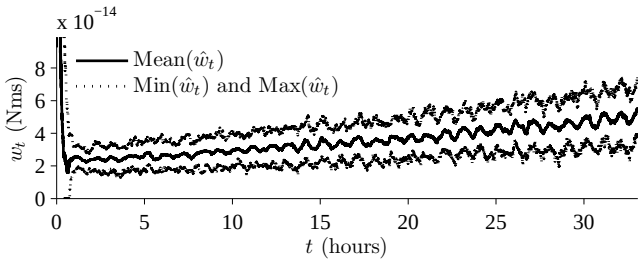
Table 2. Prognostics Performance for Thrust Bearing Wear

The prognostics performance results for thrust bearing wear using different damage models and different levels of process noise variance are shown in Table 2. Results for radial bearing wear are similar, since the same damage models were used, and are omitted here. For the thrust bearing wear, only the case using the correct damage model obtains reasonable accuracy. The estimation results for some of the damage models are shown in Fig. 8. In all cases, the damage variable, r_t, was tracked well. With the algorithm using the reference damage model, the wear parameter is tracked well and after convergence remains approximately constant. In contrast, the linear model does not capture the relationship with ω correctly (i.e., in the reference model it is really a function of ω^2), so as ω changes between the two RPM levels, the estimate of the wear parameter must constantly increase and decrease to correctly track the damage progression. Further, because the rate of damage progression in the reference model increases over time (since it is a function of r_t), and the linear model does not capture that behavior, its wear parameter estimate must increase over time. With the polynomial model also, the parameter estimates do not take on constant values. This is also due partly to the fact that a wide number of pairs of w_{t1} and w_{t2}, i.e., multiple solutions to the damage progression equation, can track the short-term damage progression well. Hence, the wear parameter estimates can change over the long-term while still tracking short-term, leading also to an increased variability in the prediction accuracy.

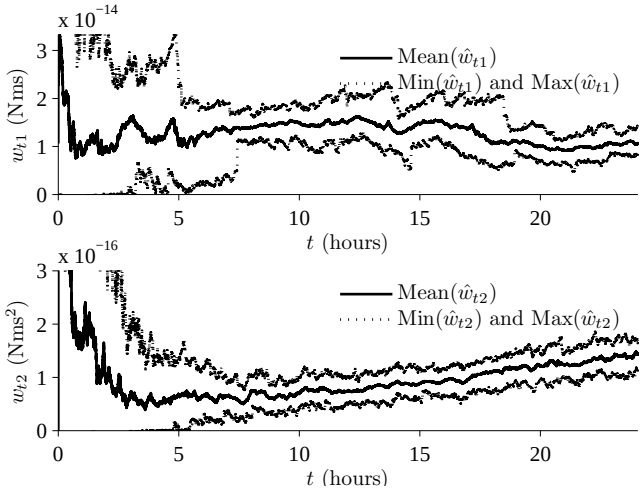
The prediction performance is compared in Fig. 9. The algorithm using the reference model obtains accurate predictions. On the other hand, the other models consistently overestimate the RUL, because at any point in time they are underesti-



(a) w_t estimation performance for $\mathcal{M}$.



(b) w_t estimation performance for $\mathcal{M}_{Linear}$.



(c) w_{t1} and w_{t2} estimation performance for $\mathcal{M}_{Poly}$.

Figure 8. Thrust bearing wear parameter estimation.

mating the rate of damage progression that would occur in the future. So, early on, the predictions are overly optimistic and could result in poor decisions based on that information. These models also produce very similar predictions. For the reference model $\mathcal{M}$, PH is around 95%, but for the remaining models, PH is around 30% or worse, so, for these models, accurate predictions are only being obtained with less than 30% life remaining, as observed in Fig. 9.

Note also that as the process noise increased, the algorithm using the reference model had only small decreases in performance, whereas for the other models, performance decreased quite significantly. In this case it was more difficult for the particle filters using these models to track damage over the

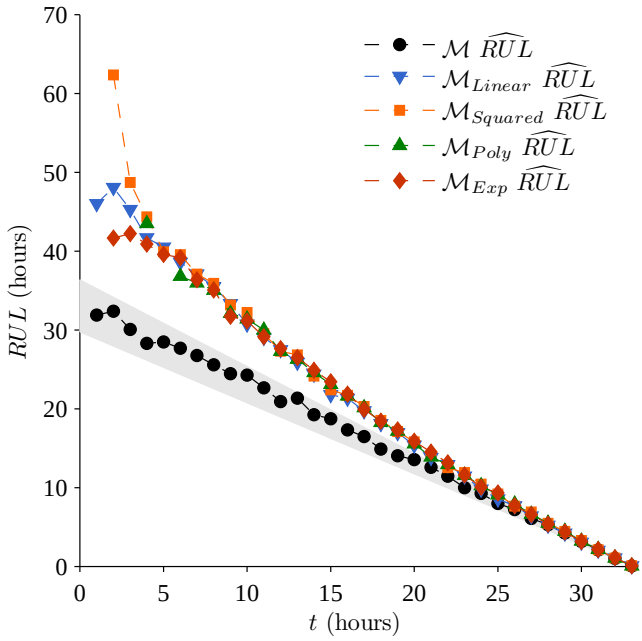


Figure 9. Thrust bearing RUL prediction performance.

short term, which resulted in a greater variation in the wear parameter estimates, leading to large decreases in accuracy.

Overall, this analysis illustrates the trade-off in the development of models of damage progression. In some cases, simple, more abstract or less granular models may suffice, especially if the system load remains constant. But with more complex operational scenarios, the need for a damage model that accurately captures the relationship with the load is necessary. In the case of the thrust bearing wear, even though the current and future inputs were the same, the fact that all of the less granular models did not account for the relationship between $\dot{r}_t$ and r_t, which caused the damage progression rate to increase over time, resulted in poor prognostics performance, even for the more complex models. The more complex models, i.e., those with more unknown wear parameters, allowed more flexibility to correctly approximate the correct damage progression function, but this also increased the dimension of the joint state-parameter space and made estimation more difficult.

8. Conclusions

We presented a model-based prognostics methodology, and investigated the effect of the choice of damage progression models on prognostics performance. In prognostics modeling, accurate damage progression models are crucial to achieving useful predictions. Using a centrifugal pump as a simulation-based case study, we developed several different damage progression models, and, assuming some physics-based wear equations as the reference form, compared the performance of the prognostics algorithm using the different

models. In some cases, such as under cyclic or constant loading, it was shown that simple linear models may suffice. Some models also performed poorly early on but achieved accurate predictions before 50% life remaining. But, omitting additional interactions within the damage progression models may cause inaccurate results, even under simple loading scenarios. Further, even though the prognostics algorithm was robust enough to track the damage with all the different models, this did not translate to accurate predictions when a different damage progression model was used relative to the reference model.

In future work, we will extend this analysis to other domains such as electrochemical systems and electrical devices, in order to establish general design guidelines for prognostics models. For a desired level of prognostics performance, we want to be able to determine what level of model granularity is necessary. These ideas also apply to data-driven models, and models for diagnosis, which will be addressed in future work as well.

ACKNOWLEDGMENTS

The funding for this work was provided by the NASA System-wide Safety and Assurance Technologies Project (SSAT) project.

REFERENCES

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Biswas, G., & Mahadevan, S. (2007, March). A Hierarchical Model-based approach to Systems Health Management. In *Proceedings of the 2007 IEEE Aerospace Conference.*

Daigle, M., & Goebel, K. (2010, March). Model-based prognostics under limited sensing. In *Proceedings of the 2010 IEEE Aerospace Conference.*

Daigle, M., & Goebel, K. (2011, March). Multiple damage progression paths in model-based prognostics. In *Proceedings of the 2011 IEEE Aerospace Conference.*

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*, 197–208.

Frantz, F. (1995). A taxonomy of model abstraction techniques. In *Proceedings of the 27th conference on Winter Simulation* (pp. 1413–1420).

Hutchings, I. M. (1992). *Tribology: friction and wear of engineering materials*. CRC Press.

Kallesøe, C. (2005). *Fault detection and isolation in centrifugal pumps*. Unpublished doctoral dissertation, Aalborg University.

Lee, K., & Fishwick, P. A. (1996). Dynamic model abstraction. In *Proceedings of the 28th conference on Winter Simulation* (pp. 764–771).

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Lyshevski, S. E. (1999). *Electromechanical Systems, Electric Machines, and Applied Mechatronics*. CRC.

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2009.*

Saha, B., Quach, P., & Goebel, K. (2011, September). Exploring the model design space for battery health management. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011.*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*.

Tu, F., Ghoshal, S., Luo, J., Biswas, G., Mahadevan, S., Jaw, L., et al. (2007, March). PHM integration with maintenance and inventory management systems. In *Proceedings of the 2007 IEEE Aerospace Conference.*

Wolfram, A., Fussel, D., Brune, T., & Isermann, R. (2001). Component-based multi-model approach for fault detection and diagnosis of a centrifugal pump. In *Proceedings of the 2001 American Control Conference* (Vol. 6, pp. 4443–4448).

Zeigler, B., Praehofer, H., & Kim, T. (2000). *Theory of modeling and simulation* (2nd ed.). Academic Press.